

Tutorial para construir un Hola Mundo en Python + Django

TheBigBro

Tutorial paso a paso

24 de octubre de 2025

1. Introducción

El objetivo de este tutorial es construir una aplicación web básica utilizando Python y Django, conectada a una base de datos MySQL en Docker.

El resultado final será una aplicación que muestre un mensaje de “Hola Mundo” con información obtenida desde la base de datos, pasando por las capas de modelo, vista y respuesta web.

2. Lenguaje, framework y ORM

- **Lenguaje:** Python 3.11+
- **Framework web:** Django 5.x
- **ORM:** Django ORM (incluido en Django, compatible con MySQL)
- **Frontend mínimo:** Plantilla HTML con Bootstrap (sin build tools adicionales)
- **Base de Datos:** MySQL ejecutándose en contenedor Docker
- **Sistema Operativo:** Windows 11

3. Preparación del Entorno

3.1 Crear el proyecto

1. Abre PowerShell o CMD.
2. Crea una carpeta para el proyecto:

```
Shell
mkdir django_mysql_app
cd django_mysql_app
```

3.2 Crear y activar el entorno virtual

```
Shell
python -m venv venv
venv\Scripts\activate
```

3.3 Instalar dependencias

```
Shell
pip install django mysqlclient
```

3.4 Crear el proyecto Django

```
Shell
django-admin startproject core .
python manage.py startapp personas
```

4. Configuración de la Base de Datos (Docker)

4.1 Crear el archivo docker-compose.yml

Crea un archivo con el siguiente contenido:

```
None
version: '3.8'

services:
  db:
    image: mysql:8.0
    container_name: mysql_django
    restart: always
    environment:
      MYSQL_DATABASE: django_db
      MYSQL_USER: django_user
      MYSQL_PASSWORD: django_pass
      MYSQL_ROOT_PASSWORD: rootpass
    ports:
      - "3306:3306"
    volumes:
```

```
- mysql_data:/var/lib/mysql

volumes:
  mysql_data:
```

4.2 Iniciar el contenedor

Ejecuta en la terminal:

```
Shell
docker-compose up -d
```

Verifica que el contenedor esté corriendo:

```
Shell
docker ps
```

5. Configuración del Proyecto Django

5.1 Conexión con la base de datos

Abre `core/settings.py` y reemplaza la configuración de DATABASES por:

```
Python
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'django_db',
        'USER': 'django_user',
        'PASSWORD': 'django_pass',
        'HOST': '127.0.0.1',
        'PORT': '3306',
    }
}
```

5.2 Registrar la aplicación

En la misma configuración (INSTALLED_APPS), agrega la aplicación personas:

```
Python
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'personas',
]
```

6. Creación del Modelo de Datos

Edita personas/models.py y define la entidad Persona:

```
Python
from django.db import models

class Persona(models.Model):
    nombre = models.CharField(max_length=100)
    edad = models.IntegerField()

    def __str__(self):
        return f"{self.nombre} ({self.edad} años)"
```

Guarda los cambios y ejecuta:

```
Shell
python manage.py makemigrations
python manage.py migrate
```

7. Cargar Datos de Prueba

Abre el shell de Django:

```
Shell  
python manage.py shell
```

Ejecuta el siguiente código:

```
Python  
from personas.models import Persona  
Persona.objects.create(nombre="Juan Pérez", edad=30)  
exit()
```

8. Implementación de la Vista “Hola Mundo”

Edita personas/views.py:

```
Python  
from django.http import HttpResponse  
from .models import Persona  
  
def hola_mundo(request):  
    persona = Persona.objects.first()  
    if persona:  
        return HttpResponse(f"¡Hola {persona.nombre}! Edad: {persona.edad}")  
    else:  
        return HttpResponse("No hay personas registradas en la base de  
datos.")
```

Edita core/urls.py para agregar la ruta principal:

```
Python  
from django.contrib import admin  
from django.urls import path  
from personas.views import hola_mundo  
  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', hola_mundo, name='hola_mundo'),  
]
```

9. Ejecución del Proyecto

Ejecuta el servidor de desarrollo:

```
Shell  
python manage.py runserver
```

Abre tu navegador y accede a:

<http://127.0.0.1:8000/>

Deberías ver el mensaje:

```
CSS  
¡Hola Juan Pérez! Edad: 30
```

10. Flujo de Funcionamiento

Django ORM se conecta a la base de datos MySQL configurada en settings.py.

Se consulta la entidad Persona mediante `Persona.objects.first()`.

La vista `hola_mundo()` procesa la información.

Django devuelve una respuesta HTTP con el contenido dinámico.